Reconstructing Noisy Polynomial Evaluation in Residue Rings

Simon R. Blackburn
Department of Mathematics, Royal Holloway, University of London
Egham, Surrey, TW20 0EX, UK
s.blackburn@rhul.ac.uk

Domingo Gomez-Perez
Faculty of Science, University of Cantabria
E-39071 Santander, Spain
gomezd@unican.es

Jaime Gutierrez
Faculty of Science, University of Cantabria
E-39071 Santander, Spain
jaime.gutierrez@unican.es

Igor E. Shparlinski
Department of Computing, Macquarie University
Sydney, NSW 2109, Australia
igor@comp.mq.edu.au

July 3, 2012

Abstract

Let $q > 1$ be an integer and let a and b be elements of the residue ring $\mathbb{Z}_q$ of integers modulo q. We show how, when given a polynomial $f \in \mathbb{Z}_q[X]$ and approximations to $v_0, v_1 \in \mathbb{Z}_q$ such that $v_1 \equiv f(v_0) \bmod q$ one can recover v_0 and v_1 efficiently. This result has direct applications to predicting the polynomial congruential generator: a sequence (v_n) of pseudorandom numbers defined by the relation $v_{n+1} \equiv f(v_n) \bmod q$ for some polynomial $f \in \mathbb{Z}_q[X]$. The applications lead to analogues of results known for the linear congruential generator $x_{n+1} \equiv ax_n + b \bmod q$, although the results are much more restrictive due to nonlinearity of the problem.

Keywords: Noisy interpolation, lattice basis reduction, polynomial congruences, predicting pseudorandom generators

1 Introduction

For an integer $q > 1$ we denote by $\mathbb{Z}_q$ the residue ring of integers modulo q. We always represent the residue classes from $\mathbb{Z}_q$ by elements of the set $\{0, 1, \ldots, q-1\}$. As usual, we denote by $\mathbb{Z}_q^*$ the set of invertible elements of $\mathbb{Z}_q$.

Accordingly, for a prime p, we denote by $\mathbb{F}_p \cong \mathbb{Z}_p$ the field of p elements and as before, we assume that it is represented by the set $\{0, 1, \ldots, p-1\}$. In particular, sometimes, where obvious, we treat elements of $\mathbb{Z}_q$ and $\mathbb{F}_p$ as integers in the above range.

Here we consider the *noisy polynomial evaluation problem* in $\mathbb{Z}_q$: given a polynomial $f(X) \in \mathbb{Z}_q[X]$ and approximations to $v_0, v_1 \in \mathbb{Z}_q$, where $v_1 \equiv f(v_0) \bmod q$, recover v_0 and v_1. By an approximation to an integer v_i, we mean an integer w_i such that $|w_i - v_i|$ is small.

The question has applications to, and has been motivated by, the predictability problem for non-linear pseudorandom number generators. To be more precise, given a polynomial $f(X) \in \mathbb{Z}_q[X]$, we define the *polynomial congruential generator* to be a sequence (v_n) of elements of $\mathbb{Z}_q$ satisfying the recurrence relation

$$v_{n+1} \equiv f(v_n) \bmod q, \qquad n = 0, 1, \ldots, \tag{1}$$

where v_0 is the *initial value*. If $\deg f = m$ then we say that the polynomial congruential generator is *of degree* m.

This generator exhibits very attractive uniformity of distribution and nonlinearity properties, see [**?**, **?**] for surveys or recent results. Here we study some of its cryptographic properties, namely the question of so-called *predictability* of such generators.

In the cryptographic setting, the initial value v_0 (and sometimes the polynomial f and the modulus q) is assumed to be secret, and we want to use the output of the generator as a stream cipher. In this setting, we output only the most significant bits of each v_n in the hope that this makes the resulting output sequence difficult to predict. (Note that if we remove the k least significant bits of each v_n, an evesdropper may easily find integers w_n such that $|w_n - v_n| \le 2^{k-1}$ by examining the output. This is the connection to the noisy polynomial evaluation problem.) The main result of this paper may be interpreted as saying that if f and q are public, and if too many bits of the elements v_n are output at each stage, then the generator becomes insecure because the elements v_n may be efficiently recovered from the output. Slightly more precisely, we show that the polynomial congruential generator is polynomial time (in $\log q$ and $\deg f$) predictable if sufficiently many bits of its consecutive elements are revealed (even if the degree of the generator is allowed to slowly grow together with the size of the modulus q). Our results exclude a small set of polynomials f, and a small set of starting values v_0: see Theorems **??** and **??** for the details. In the final section of the paper, we discuss the case when the polynomial f forms part of the secret key, and show that the unique recovery of the elements v_n from the output is not possible.

For the *linear congruential generator*

$$x_{n+1} \equiv ax_n + b \bmod q, \qquad n = 0, 1, \ldots, \tag{2}$$

similar problems have been introduced by Knuth [**?**] and then considered in [**?**, **?**, **?**, **?**, **?**]; see also surveys [**?**, **?**]. We remark that predicting nonlinear generators has been considered in some of these works as well, however only in the case when all terms are given in full. Thus the case we consider here, when only some bits of the output are given, has not previously been studied for non-linear generators.

Several nonlinear generators have recently been studied in [**?**, **?**]. Here, as in [**?**, **?**], we use some lattice algorithms. However, in contrast to [**?**, **?**], the dimension of our lattices grows as $\deg f$ grows, and thus slightly different tools need to be applied.

In some sense the problem we solve can be considered as a special case of the problem of finding small solutions of multivariate polynomial congru-

ences. For polynomial congruences in one variable, an algorithm for solving this problem has been given by Coppersmith [**?**], see also [**?**, **?**]. However, in the general case only heuristic results are known. Here we are able to obtain rigorous results, due to the special structure of the polynomials involved.

Throughout the paper, the constants in the 'O'-notation are absolute.

Acknowledgment. The authors would like to thank Harald Niederreiter for his interest and helpful discussions. This paper was written during visits of I.S. to the University of Cantabria (supported by MECD grant SAB2000-0260) and to Royal Holloway, University of London (supported by an EPSRC Visiting Fellowship). D.G.-P. and J.G. were partially supported by Spanish Ministry of Science grant BFM2001-1294. The support and hospitality of all these organisations are gratefully acknowledged.

2 Preliminaries

2.1 Background on Lattices

Here we review several related results and definitions concerning lattices, all of which can be found in [**?**]. For more details and more recent references, we recommend consulting [**?**, **?**, **?**, **?**, **?**].

Let $\{\mathbf{b}_1, \ldots, \mathbf{b}_s\}$ be a set of linearly independent vectors in $\mathbb{R}^r$. The set

$$\mathcal{L} = \{\mathbf{z} \ : \ \mathbf{z} = c_1\mathbf{b}_1 + \ldots + c_s\mathbf{b}_s, \quad c_1, \ldots, c_s \in \mathbb{Z}\}$$

is called an *s-dimensional lattice* with *basis* $\{\mathbf{b}_1, \ldots, \mathbf{b}_s\}$. If $s = r$, the lattice L is of *full rank*.

One basic lattice problem is the *shortest vector problem*: given a basis of a lattice $\mathcal{L}$ in $\mathbb{R}^s$, find a nonzero lattice vector $\mathbf{f} \in \mathcal{L}$ which minimises the Euclidean norm $\|\mathbf{f}\|$ among all lattice vectors. Unfortunately, there are several indications that this problem is **NP**-complete (when the dimension grows). In particular, it is shown in [**?**] that the shortest vector problem is **NP**-hard under randomized reductions, and so it is now widely believed that there is no polynomial time algorithm to solve SVP. For a slightly weaker task of finding a short vector, the celebrated *LLL algorithm* of Lenstra, Lenstra and Lovász [**?**] provides a desirable solution. We however use a slightly stronger result which follows from [**?**], and which we state as Lemma **??**.

We always assume that the basis of $\mathcal{L}$ consists of vectors with rational components. Thus a polynomial time algorithm for $\mathcal{L}$ means an algorithm

whose running time is polynomial in the total number of bits required for binary representation of numerators and denominators of all components of the basis.

Lemma 1. *There exists a deterministic polynomial time algorithm which, when given an s-dimensional full rank lattice $\mathcal{L}$, finds a non-zero lattice vector $\mathbf{f} \in \mathcal{L}$ satisfying the inequality*

$$\|\mathbf{f}\| \le \lambda_s \min\{\|\mathbf{z}\| : \quad \mathbf{z} \in \mathcal{L},\ \mathbf{z} \ne \mathbf{0}\},$$

where

$$\lambda_s = \exp\left(O\left(\frac{s(\log\log s)^2}{\log s}\right)\right).$$

Many other results on both exact and approximate finding of a shortest vector in a lattice are discussed in [**?**, **?**, **?**, **?**, **?**, **?**]; see also [**?**] for the most recent developments (which however lead to probabilistic algorithms).

In fact, in this paper we consider only very special lattices. Namely, we consider only lattices which consist of integer solutions $\mathbf{x} = (x_0, \ldots, x_{s-1}) \in \mathbb{Z}^s$ of the system of congruences

$$\sum_{i=0}^{s-1} a_{ij} x_i \equiv 0 \bmod q_i, \qquad j = 1, \ldots, \ell,$$

modulo some integers $q_1, \ldots, q_\ell$. The lattices we consider are full rank lattices of dimension s. All the aforementioned algorithms become polynomial in $\log(q_1 \ldots q_\ell)$ when applied to such lattices.

2.2 Polynomial Congruences

Our second basic tool is an upper bound on the number of solutions of polynomial congruences.

For congruences modulo a prime we can use the *Lagrange theorem* which asserts that a non-zero polynomial of degree s over any field has no more than s zeros in this field.

However for congruences modulo composite numbers we apply an upper bound from [**?**].

For a polynomial

$$F(X) = \sum_{i=0}^{s} A_i X^i \in \mathbb{Z}[X] \tag{3}$$

of degree s and an integer $Q \geq 1$ we denote by $T(F, Q)$ the number of solutions of the congruence

$$F(x) \equiv 0 \bmod Q, \qquad x \in \mathbb{Z}_Q.$$

We now define $N_s(Q)$ as the largest possible value of $T(F, Q)$ taken over all polynomials (**??**) with $\gcd(A_0, \ldots, A_s, Q) = 1$. (Note that there is no restriction on A_0.)

The following bound is a relaxed form of the main result of [**?**].

Lemma 2. *The bound*

$$N_s(Q) = O(sQ^{1-1/s})$$

holds.

Writing $F(X) = DG(X)$ with $D = \gcd(A_0, \ldots, A_s, Q)$ and $G(X) \in \mathbb{Z}[X]$, we also have that $T(F, Q) \leq DN_s(Q/D)$ for any polynomial (**??**), so

$$T(F, Q) = O(sQ^{1-1/s}D^{1/s}). \tag{4}$$

We apply the Lagrange theorem and Lemma **??** to some families of polynomials parametrised by small vectors in a certain lattice, thus the size of the family can be kept under control. Zeros of these polynomials correspond to potentially "bad" initial values of the polynomial congruential generator (**??**). Thus, if all polynomials in this family are not identical to zero modulo q (or to be more precise, have a not too large value of D in (**??**)) then we have an upper bound on the number of such "bad" initial values. Hence, the most crucial part of our approach is to study possible vanishing of polynomials in the above family and to show that this may happen only for very few values of the coefficients of the generator (**??**).

2.3 Residues of Small-Height Fractions

Some exceptional sets of parameters in our results can be described as sets of residues of fractions with bounded numerator and denominator. Namely, let $\mathcal{F}(q, R, S)$ be the set of $a \in \mathbb{Z}_q^*$ that satisfy a congruence of the form

$ar \equiv s \bmod q$ for some integers r and s, not both zero modulo q, where $|r| \le R$ and $|s| \le S$.

As usual, we use $\sigma(q)$ to denote the sum of divisors of q.

Lemma 3. *For any $1 \le R, S < q$, the bound*

$$\#\mathcal{F}(q, R, S) \le 4RS\frac{\sigma(q)}{q}$$

holds.

Proof. For every $a \in \mathbb{Z}_q^*$, the congruence $ar \equiv s \bmod q$ implies

$$\gcd(r, q) \mid \gcd(s, q).$$

We count the elements of $\mathcal{F}(q, R, S)$ by first choosing a divisor $d < q$ of q, then choosing r and s such that $|r| \le R$, $|s| \le S$, $\gcd(r, q) = d$ and $d|s$, and finally choosing a such that $ar \equiv s \bmod q$. Note that once d is chosen, there are at most $2R/d$ choices for r and at most $2S/d$ choices for s (because $1 \le R, S < q$ we see that $rs \ne 0$). Moreover, once r and s such that $\gcd(r, q) = d$ are fixed, there are at most d choices for a. Hence

$$\#\mathcal{F}(q, R, S) \le \sum_{d|q} \frac{2R}{d} \cdot \frac{2S}{d} \cdot d \le 4RS \sum_{d|q} \frac{1}{d} = 4RS\frac{\sigma(q)}{q}$$

which finishes the proof. □

Recall that

$$\sigma(q) = O\left(q \log\log q\right);$$

see [**?**, Theorem 323]. In particular,

$$\#\mathcal{F}(q, R, S) = O\left(RS \log\log q\right).$$

3 Main Results

3.1 Predicting the Polynomial Generator Modulo an Arbitrary Integer

Let Δ be a positive integer. We say an integer w is a Δ-*approximation* to an integer v if $|w - v| \le \Delta$.

Recall that we use $\sigma(q)$ to denote the sum of the divisors of an integer q, and we define λ_s to be the "stretch" factor λ_s given in Lemma **??**.

We are now ready to state the main theorem of the paper.

Theorem 4. *There exists an algorithm with the following properties. Let q and Δ be integers such that $q > \Delta \geq 1$ and $\gcd(q, \Delta) = 1$. Let*

$$f(X) = \sum_{i=0}^{m} a_i X^i \in \mathbb{Z}_q[X]$$

be a polynomial of degree $m \geq 2$ over $\mathbb{Z}_q$ whose leading coefficient a_m lies in $\mathbb{Z}_q^* \backslash \mathcal{A}_m(q, \Delta)$, for some set $\mathcal{A}_m(q, \Delta)$ of cardinality at most

$$16(m+2)\lambda_{m+2}^2 \Delta^{m+1} \frac{\sigma(q)}{q}.$$

The algorithm, when given f and Δ-approximations w_0, w_1 to v_0, v_1 where $v_1 \equiv f(v_0) \bmod q$, recovers v_0, v_1 in time polynomial in m and $\log q$ provided that v_0 does not lie in a certain set $\mathcal{V}(f) \subseteq \mathbb{Z}_q$ of cardinality $\#\mathcal{V}(f) = O\left((2\Delta)^{\vartheta_m} q^{1-1/(m-1)}\right)$, where

$$\vartheta_m = \frac{m^3 + 3m - 2}{2(m-1)}.$$

Proof. We may assume that

$$q > 2^{m+1}\sqrt{m+2}\,\lambda_{m+2}\Delta^{m-1} \qquad \text{and} \qquad q > \Delta^m \tag{5}$$

for if either of these inequalities fail to hold the result is trivially true (by examining bound on the cardinality of the set $\mathcal{A}_m(q, \Delta)$).

We define the set $\mathcal{A}_m(q, \Delta) = \mathcal{F}(q, R, S)$, with $R = 2\sqrt{m+2}\,\lambda_{m+2}\Delta^m$ and $S = 2\sqrt{m+2}\,\lambda_{m+2}\Delta$, where $\mathcal{F}(q, R, S)$ is defined in Section **??**. By (**??**) we see that $R < S < q$. Now the bound on $\#\mathcal{A}_m(q, \Delta)$ is immediate by Lemma **??**.

An outline of the algorithm is as follows. The algorithm first constructs a lattice $\mathcal{L}$ from the information it is given. This lattice has a short non-zero vector $\mathbf{e}$ which may be used to derive v_0 and v_1 from w_0 and w_1. The lattice $\mathcal{L}$ has the additional property that any reasonably short vector in $\mathcal{L}$ is parallel to $\mathbf{e}$. It is also important to observe that the bit-size of all coordinates of

the basis vectors of $\mathcal{L}$ is $O(\log q)$. The algorithm finds a reasonably short non-zero vector $\mathbf{f} \in \mathcal{L}$ by using the techniques of Lemma **??**. It is then easy to find $\mathbf{e}$ and hence v_0 and v_1.

Let $\varepsilon_j = v_j - w_j$, $j = 0, 1$. Then we have

$$w_1 + \varepsilon_1 \equiv \sum_{i=0}^{m} a_i (w_0 + \varepsilon_0)^i \bmod q.$$

If we expand the right hand side of this equation in terms of powers of ϵ_0 using Taylor's formula, and then introduce various powers of Δ that cancel each other, we obtain

$$A\Delta^m + B\Delta^{m-1}\varepsilon_1 + \sum_{i=1}^{m} C_i \Delta^{m-i} \varepsilon_0^i \equiv 0 \bmod q,$$

where

$$\begin{aligned} A &\equiv (f(w_0) - w_1)\Delta^{-m} \bmod q, \\ B &\equiv -\Delta^{-m+1} \bmod q, \\ C_i &\equiv \frac{f^{(i)}(w_0)}{i!}\Delta^{-m+i} \bmod q, \quad i = 1, \ldots, m, \end{aligned}$$

and $f^{(i)}$ denotes the ith derivative of f.

Let $\mathcal{L}$ be the lattice consisting of integer solutions $\mathbf{x} = (x_0, \ldots, x_{m+1}) \in \mathbb{Z}^{m+2}$ of the system of congruences:

$$\begin{aligned} Ax_0 + Bx_1 + \sum_{i=1}^{m} C_i x_{i+1} &\equiv 0 \bmod q, \\ x_0 &\equiv 0 \bmod \Delta^m, \\ x_1 &\equiv 0 \bmod \Delta^{m-1}, \\ x_{i+2} &\equiv 0 \bmod \Delta^{m-i-1}, \qquad i = 0, \ldots, m-1. \end{aligned} \tag{6}$$

Note that $\mathcal{L}$ can be computed from the information given to the algorithm and in fact it is easy to see that it is a simple linear algebra problem to compute a basis of $\mathcal{L}$ whose basis vectors consist of elements of bit-size $O(\log q)$.

Clearly, $\mathcal{L}$ contains the non-zero vector

$$\begin{aligned} \mathbf{e} &= (\Delta^m, \Delta^{m-1}\varepsilon_1, \Delta^{m-1}\varepsilon_0, \ldots, \Delta^{m-i}\varepsilon_0^i, \ldots, \varepsilon_0^m) \\ &= (\Delta^m e_0, \Delta^{m-1} e_1, \Delta^{m-1} e_2, \ldots, \Delta^{m-i+1} e_i, \ldots, e_{m+1}). \end{aligned}$$

We have

$$e_0 = 1, \quad |e_1| \le \Delta, \qquad |e_i| \le \Delta^{i-1}, \qquad i = 2, \ldots, m+1.$$

Since $\Delta \ge 2$ and $m \ge 2$, we see that the Euclidean norm $\|\mathbf{e}\|$ of $\mathbf{e}$ satisfies the inequality

$$\|\mathbf{e}\| \ \le \ \sqrt{(m+2)\Delta^{2m}} = \sqrt{m+2}\,\Delta^m.$$

The algorithm of Lemma **??** applied to the lattice $\mathcal{L}$ returns a non-zero vector

$$\mathbf{f} = (\Delta^m f_0, \Delta^{m-1} f_1, \Delta^{m-1} f_2, \ldots, \Delta^{m-i+1} f_i, \ldots, f_{m+1}) \in \mathcal{L}$$

such that $\|\mathbf{f}\| \le \lambda_{m+2}\|\mathbf{e}\| \le \sqrt{m+2}\lambda_{m+2}\Delta^m$. In particular, we have the inequalities

$$\begin{aligned} &|f_0| \le \sqrt{m+2}\,\lambda_{m+2}, \qquad |f_1| \le \sqrt{m+2}\,\lambda_{m+2}\Delta, \\ &|f_i| \le \sqrt{m+2}\,\lambda_{m+2}\Delta^{i-1}, \quad i = 2, \ldots, m+1. \end{aligned}$$

We aim to show that $\mathbf{f}$ is parallel to $\mathbf{e}$, provided that v_0 does not lie in a set $\mathcal{V}(f)$ which we define below.

The vector $f_0\mathbf{e} - e_0\mathbf{f} \in \mathcal{L}$ has first component zero, and so using the first congruence in (**??**) we obtain

$$B\Delta^{m-1}d_1 + \sum_{i=1}^{m} C_i \Delta^{m-i} d_{i+1} \equiv 0 \bmod q,$$

where $d_i = f_0 e_i - e_0 f_i = f_0 e_i - f_i$, $i = 1, \ldots, m+1$. Hence

$$\begin{aligned} &|d_1| \le 2\sqrt{m+2}\,\lambda_{m+2}\Delta, \\ &|d_i| \le 2\sqrt{m+2}\,\lambda_{m+2}\Delta^{i-1}, \qquad i = \ 2, \ldots, m+1. \end{aligned} \tag{7}$$

Using the definitions of B and $C_1, \ldots, C_m$ (and the fact that C_m is equal to the leading coefficient a_m of $f(X)$) we have

$$\sum_{i=1}^{m-1} \frac{f^{(i)}(w_0)}{i!} d_{i+1} \equiv d_1 - a_m d_{m+1} \bmod q. \tag{8}$$

We remark that if $d_2 \equiv \ldots \equiv d_m \equiv 0 \bmod q$, then (**??**) implies that $d_1 \equiv a_m d_{m+1} \bmod q$. Recalling that $a_m \in \mathbb{Z}_q^* \backslash \mathcal{A}_m(q, \Delta)$ we then derive that

$d_1 \equiv d_{m+1} \equiv 0 \bmod q$. Taking into account the bound (**??**) we conclude that in this case $d_i = 0$, $i = 1, \ldots, m+1$. Hence $f_0\mathbf{e} - e_0\mathbf{f} = 0$, and so $\mathbf{f}$ and $\mathbf{e}$ are parallel. Hence we may assume that one of $d_2, d_3, \ldots, d_m$ is non-zero modulo q.

Substituting $w_0 = v_0 - \varepsilon_0$ in the congruence (**??**), we obtain the congruence

$$F(v_0) \equiv \alpha_0 \bmod q, \tag{9}$$

where

$$F(X) = \sum_{i=1}^{m-1} \alpha_i X^i$$

and α_i, $i = 0, \ldots, m-1$, are polynomials in $\varepsilon_0, d_1, \ldots, d_{m+1}$. We place any solution v_0 to (**??**) for any possible values of $d_1, \ldots, d_{m+1}$ and ε_0 into the set $\mathcal{V}(f)$. Thus $\mathbf{e}$ and $\mathbf{f}$ are parallel, so long as $v_0 \notin \mathcal{V}(f)$. We need to show that the cardinality of $\mathcal{V}(f)$ is as claimed in the statement of the theorem.

We define ν by the conditions $d_2 = \ldots = d_\nu = 0$, $d_{\nu+1} \neq 0$. We are assuming that not all of $d_2, \ldots, d_m$ are zero, and so $\nu \leq m-1$. Then F is of degree $\deg F = \deg f^{(\nu)} = m - \nu$ and the leading coefficient of F is

$$\alpha_{m-\nu} = \binom{m}{\nu} a_m d_{\nu+1}.$$

Note that this coefficient is non-zero modulo q since $a_m \in \mathbb{Z}_q^*$ and that by (**??**)

$$\left| \binom{m}{\nu} d_{\nu+1} \right| \leq 2^m d_{\nu+1} \leq 2^{m+1}\sqrt{m+2}\,\lambda_{m+2}\Delta^\nu < q$$

by our assumption (**??**). Moreover we see that

$$\begin{aligned} \gcd(\alpha_1, \ldots, \alpha_{m-1}, q) &\leq \gcd(\alpha_{m-\nu}, q) = \gcd\left(\binom{m}{\nu} d_{\nu+1}, q \right) \\ &\leq \left| \binom{m}{\nu} d_{\nu+1} \right| \leq 2^{m+1}\sqrt{m+2}\,\lambda_{m+2}\Delta^\nu. \end{aligned}$$

Thus by (**??**) we see that each congruence (**??**) can be satisfied by at most

$$O\left((m-\nu) q^{1-1/(m-\nu)} (2^{m+1}\sqrt{m+2}\,\lambda_{m+2}\Delta^\nu)^{1/(m-\nu)} \right)$$

values v_0. Note that, for $1 \le \nu \le m-1$,

$$\begin{aligned}(m-\nu)q^{1-1/(m-\nu)}(2^{m+1}\sqrt{m+2}\lambda_{m+2}\Delta^{\nu})^{1/(m-\nu)} &= O\left(m2^m\lambda_{m+2}q^{1-1/(m-\nu)}\Delta^{\nu/(m-\nu)}\right)\\ &= O\left(m2^m\lambda_{m+2}q\Delta^{-1}(\Delta^m/q)^{1/(m-\nu)}\right)\\ &= O\left(m2^m\lambda_{m+2}q\Delta^{-1}(\Delta^m/q)^{1/(m-1)}\right),\end{aligned}$$

where the last equality follows from (**??**).

Thus we have placed at most $O\left(m2^m\lambda_{m+2}q^{1-1/(m-1)}\Delta^{1/(m-1)}\right)$ values of v_0 in $\mathcal{V}(f)$ for each choice of $\varepsilon_0, d_1, \ldots, d_{m+1}$. By (**??**) the total number of possible choices for the integers d_i, $i = 1, \ldots, m+1$, is at most

$$\left(4\sqrt{m+2}\lambda_{m+2}\Delta+1\right)\prod_{i=2}^{m+1}\left(4\sqrt{m+2}\lambda_{m+2}\Delta^{i-1}+1\right) < \left(5\sqrt{m+2}\lambda_{m+2}\right)^{m+1}\Delta^{m(m+1)/2+1}$$

and the total number of possible choices for ε_0 is at most $2\Delta+1$. Thus the total number of values of v_0 that we have placed in $\mathcal{V}(f)$ is

$$O\left(m\left(10\sqrt{m+2}\lambda_{m+2}\right)^{m+2}q^{1-1/(m-1)}\Delta^{m(m+1)/2+2+1/(m-1)}\right) = O\left((2\Delta)^{\vartheta_m}q^{1-1/(m-1)}\right),$$

where

$$\vartheta_m = \frac{m(m+1)}{2}+2+\frac{1}{m-1} = \frac{m^3+3m-2}{2(m-1)}.$$

We have shown that $\mathbf{e}$ and $\mathbf{f}$ are always parallel, for otherwise v_0 would lie in the set $\mathcal{V}(f)$ of values which we have excluded. Since $e_0 = 1$, we find that $\mathbf{e} = \mathbf{f}/f_0$ and thus the algorithm may now recover $\mathbf{e}$ from $\mathbf{f}$. Obviously, given the third component $\Delta^{m-1}\varepsilon_0$ of $\mathbf{e}$ the algorithm can find v_0. This completes the proof. □

3.2 Predicting the Polynomial Generator Modulo a Prime

Let p be a prime. Let Δ and m be integers such that $p \ge \Delta \ge 1$ and $m \ge 2$. We also use the notion of a Δ-approximation given in Section **??**.

Theorem 5. *There exists an algorithm with the following property. Let p be a prime number, and let Δ be an integer such that $p > \Delta \geq 1$. Let*

$$f(X) = \sum_{i=0}^{m} a_i X^i \in \mathbb{F}_p[X]$$

be a polynomial of degree $m \geq 2$ over $\mathbb{F}_p$ whose leading coefficient a_m lies in $\mathbb{F}_p^* \backslash \mathcal{A}_m(p, \Delta)$ for some set $\mathcal{A}_m(p, \Delta)$ of cardinality

$$\#\mathcal{A}_m(p, \Delta) < 17\lambda_{m+2}^2 (m+2)\Delta^{m+1}.$$

Then the algorithm, when given f and Δ-approximations w_0, w_1 to v_0, v_1 where $v_1 \equiv f(v_0) \bmod p$, recovers v_0, v_1 in time polynomial in m and $\log p$ provided that v_0 does not lie in a certain set $\mathcal{V}(f) \subseteq \mathbb{F}_p$ of cardinality $\#\mathcal{V}(f) = O\left((2\Delta)^{m(m+1)/2+2}\right)$.

Proof. The proof of this theorem is almost identical to that of Theorem **??**.

In particular, we define $\mathcal{A}_m(p, \Delta) = \mathcal{F}(p, R, S)$ where as before $R = 2\sqrt{m+2}\,\lambda_{m+2}\Delta^m$, $S = 2\sqrt{m+2}\,\lambda_{m+2}\Delta$ and $\mathcal{F}(p, R, S)$ is defined as in Section **??**. Again, we can assume that $2\sqrt{m+2}\,\lambda_{m+2}\Delta^m < p$, and also that $p \geq 17$, so that $\sigma(p)/p = (p+1)/p < 17/16$. Now the bound on $\#\mathcal{A}_m(p, \Delta)$ follows from Lemma **??**.

The only other place where the proof differs from that of Theorem **??** is when we calculate the cardinality of the set $\mathcal{V}(f)$; so we need to count the number of possible solutions to congruences of the form

$$F(v_0) \equiv \alpha_0 \bmod p, \tag{10}$$

where

$$F(X) = \sum_{i=1}^{m-1} \alpha_i X^i$$

and α_i, $i = 0, \ldots, m-1$, are polynomials in $\varepsilon_0, d_1, \ldots, d_{m+1}$. Just as in the proof of Theorem **??**, all these congruences are non-trivial, and so (since we are working modulo a prime) each instance of (**??**) has at most $m-1$ solutions. Moreover, as in the proof of Theorem **??** we see by (**??**) that there are at most $\left(5\sqrt{m+2}\lambda_{m+2}\right)^{m+1} \Delta^{m(m+1)/2+1}$ possibilities for $d_1, \ldots, d_{m+1}$ and at most $2\Delta + 1$ possibilities for ε_0 and hence at most

$$(m-1)(2\Delta+1)\left(5\sqrt{m+2}\lambda_{m+2}\right)^{m+1} \Delta^{m(m+1)/2+1} = O\left((2\Delta)^{m(m+1)/2+2}\right)$$

solutions to a congruence of the form (**??**). The proof of Theorem **??**, with this counting argument changed, now suffices to prove Theorem **??**. □

4 Remarks and Open Questions

It would be very natural to study the case when the polynomial f is not known and forms a part of the secret key. However, we observe that in this case the unique recovery of v_0 (and f) is not possible. Indeed, it is easy to see that given any number k of 'approximations' w_j, which are actually the exact values $w_j = v_j$, $j = 0, \ldots, k-1$, and an integer h, each of the sequences,

$$v_0^{(h)} = v_0 - h \qquad \text{and} \qquad v_j^{(h)} \equiv f_h\left(v_{j-1}^{(h)}\right) \bmod q, \quad j = 1, \ldots, k-1,$$

where $f_h(X) = f(X+h) - h$ satisfies $v_j^{(h)} = v_j - h$. Therefore, for any integer h with $|h| \le \Delta$ we have $|w_j - v_j^{(h)}| \le \Delta$. Thus each of the sequences $v_{j-1}^{(h)}$ (and each polynomial f_h) may give rise to the same sequence of approximations w_j. We remark that this argument works for any family $\mathcal{F}$ of functions which is closed under the transformation $f(X) \to f(X+h) - h$. The fact that the family of functions $f_{a,b}(X) = aX^{-1} + b$ does not satisfy this property explains why the *inversive congruential generator*, $u_{n+1} \equiv au_n^{-1} + b \bmod q$, can be completely recovered even in the case of unknown coefficients; see [**?**, **?**] for the case where $q = p$ is prime. On the other hand, in cryptographic applications we do not need to completely recover v_0 and f: we merely need to be able to continue to generate the sequence of "approximations" w_j (formed, say, by taking the $\ell > 0$ most significant bits of v_j, that is $w_j = 2^\ell \left\lfloor 2^{-\ell} v_j \right\rfloor$). In the case of the linear congruential generator (**??**), that is, for the family of functions $f(X) = aX + b$, this issue has been discussed in Section 3 of [**?**]. In particular it has been noted in [**?**] that the difference sequence $y_n = x_{n+1} - x_n$ satisfies the homogeneous relations

$$y_{n+1} \equiv ay_n \pmod q, \qquad n = 0, 1, \ldots,$$

and can be recovered, which can then be used for finding approximations to the sequence x_n. However, for nonlinear functions f this method no longer applies, and finding an analogous method (even a heuristic one) remains an open problem.

In Theorem **??** we have the technical condition that $\gcd(\Delta, q) = 1$. This condition is needed to be able to define the coefficients $A, B, C_1, \ldots, C_m$.

However, the condition is rather an artificial one: the value Δ in the algorithm of Theorem **??** may be replaced by any slightly larger value Δ_0 without significantly altering the algorithm's performance, and so we may ensure that $\gcd(\Delta_0, q) = 1$. For example, Δ_0 can be chosen to be the smallest prime number which is greater than Δ and does not divide q. Because q has at most $O(\log q/\log\log q)$ prime divisors this would lead to only slightly weaker estimates. More precisely, the largest distance $J(q)$ between two integers relatively prime to q is called the *Jacobsthal function* and has been extensivel studied in the literature, in particular $J(q) = O((\log q)^2)$, see [**?**].

We have not used the full power of the bound on λ_s in Lemma **??**. However using the original estimate $\lambda_s \le 2^{(s-1)/2}$ of [**?**] would force us to replace 2Δ in our bounds on $\mathcal{A}_m(p, \Delta)$ and $\mathcal{V}(f)$ in Sections **??** and **??** with a slightly larger multiple of Δ.

References

[1] M. Ajtai, 'The shortest vector problem in L_2 is NP-hard for randomized reductions', *Proc. 30th ACM Symp. on Theory of Comput.*, ACM, 1998, 10–19.

[2] M. Ajtai, R. Kumar and D. Sivakumar, 'A sieve algorithm for the shortest lattice vector problem', *Proc. 33rd ACM Symp. on Theory of Comput.*, ACM, 2001, 601–610.

[3] S. R. Blackburn, D. Gomez-Perez, J. Gutierrez and I. E. Shparlinski, 'Predicting the inversive generator', *Proc. 9th IMA Intern. Conf on Cryptography and Coding*, Lect. Notes in Comp. Sci., Springer-Verlag, Berlin, **2898** (2003), 264–275.

[4] S. R. Blackburn, D. Gomez-Perez, J. Gutierrez and I. E. Shparlinski, 'Predicting nonlinear pseudorandom number generators', *Math. Comp.*, (to appear).

[5] J. Boyar, 'Inferring sequences produced by pseudo-random number generators', *J. ACM*, **36** (1989), 129–141.

[6] J. Boyar, 'Inferring sequences produces by a linear congruential generator missing low–order bits', *J. Cryptology* **1** (1989) 177–184.

[7] E. F. Brickell and A. M. Odlyzko, 'Cryptanalysis: A survey of recent results', *Contemp. Cryptology*, IEEE Press, NY, 1992, 501–540.

[8] D. Coppersmith, 'Small solutions to polynomial equations, and low exponent RSA vulnerabilities', *J. Cryptology*, **10** (1997), 233–260.

[9] D. Coppersmith, 'Small solutions of small degree polynomials', *Proc. Intern. Conf. on Cryptography and Lattices*, Lect. Notes in Comp. Sci., vol. 2146, Springer-Verlag, Berlin, 2001, 20–31.

[10] A. M. Frieze, J. Håstad, R. Kannan, J. C. Lagarias and A. Shamir, 'Reconstructing truncated integer variables satisfying linear congruences', *SIAM J. Comp.*, **17** (1988), 262–280.

[11] M. Grötschel, L. Lovász and A. Schrijver, *Geometric algorithms and combinatorial optimization*, Springer-Verlag, Berlin, 1993.

[12] G. H. Hardy and E. M. Wright, *An introduction to the theory of numbers*, Oxford Univ. Press, Oxford, 1979.

[13] N. A. Howgrave-Graham, 'Finding small roots of univariate modular equations revisited', *Proc. 6th IMA Intern. Conf on Cryptography and Coding*, Lect. Notes in Comp. Sci., vol. 1355, Springer-Verlag, Berlin, 1997, 131–142.

[14] H. Iwaniec, 'On the problem of Jacobsthal', *Demonstratio Math.*, **11** (1978), 225–231.

[15] A. Joux and J. Stern, 'Lattice reduction: A toolbox for the cryptanalyst', *J. Cryptology*, **11** (1998), 161–185.

[16] R. Kannan, 'Algorithmic geometry of numbers', *Annual Review of Comp. Sci.*, **2** (1987), 231–267.

[17] R. Kannan, 'Minkowski's convex body theorem and integer programming', *Math. Oper. Res.*, **12** (1987), 415–440.

[18] D. E. Knuth, 'Deciphering a linear congruential encryption', *IEEE Trans. Inf. Theory* **31** (1985), 49–52.

[19] S. V. Konyagin, 'On the number of solutions of an univariate congruence of nth degree', *Matem. Sbornik*, **102** (1979), 171–187 (in Russian).

[20] H. Krawczyk, 'How to predict congruential generators', *J. Algorithms*, **13** (1992), 527–545.

[21] J. C. Lagarias, 'Pseudorandom number generators in cryptography and number theory', *Proc. Symp. in Appl. Math.*, Amer. Math. Soc., Providence, RI, **42** (1990), 115–143.

[22] A. K. Lenstra, H. W. Lenstra and L. Lovász, 'Factoring polynomials with rational coefficients', *Mathematische Annalen*, **261** (1982), 515–534.

[23] D. Micciancio and S. Goldwasser, *Complexity of lattice problems*, Kluwer Acad. Publ., 2002.

[24] P. Q. Nguyen and J. Stern, 'Lattice reduction in cryptology: An update', *Proc. 4th Intern. Symp. on Algorithmic Number Theory*, Lect. Notes in Comp. Sci., vol. 1838, Springer-Verlag, Berlin, 2000, 85–112.

[25] P. Q. Nguyen and J. Stern, 'The two faces of lattices in cryptology', *Proc. Intern. Conf. on Cryptography and Lattices*, Lect. Notes in Comp. Sci., vol. 2146, Springer-Verlag, Berlin, 2001, 146–180.

[26] H. Niederreiter and I. E. Shparlinski, 'Recent advances in the theory of nonlinear pseudorandom number generators', *Proc. Conf. on Monte Carlo and Quasi-Monte Carlo Methods, 2000*, Springer-Verlag, Berlin., 2002, 86–102.

[27] H. Niederreiter and I. E. Shparlinski, 'Dynamical systems generated by rational functions', *Proc. 15th Symp. on Appl. Algebra, Algebraic Algorithms, and Error-Correcting Codes*, Lect. Notes in Comp. Sci., vol. 2643, Springer-Verlag, Berlin, 2003, 6–17.

[28] C. P. Schnorr, 'A hierarchy of polynomial lattice basis reduction algorithms', *Theor. Comp. Sci.*, **53** (1987), 201–224.